DSC 140B

Representation Learning

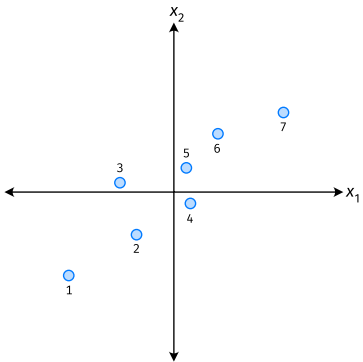
Lecture 06 | Part 1

Dimensionality Reduction

Last Time: Dimensionality Reduction

- **Given:** data points $\vec{x}^{(1)}, \ldots, \vec{x}^{(n)} \in \mathbb{R}^d$
- **Goal:** create a new, lower-dimensional data set without losing too much useful information
- For now, focus on reducing to just one dimension.

Example



- Each point is a phone.
- $\vec{x} = (\text{width}, \text{weight})^T$.
- Can we reduce $\vec{x}$ to a single feature, z, without losing too much information?

The Idea from Last Time

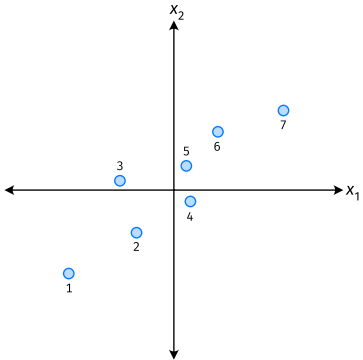
- Our new feature should be a “mixture” of the old features:

$$\begin{aligned} z &= u_1 \times \text{width} + u_2 \times \text{weight} \\ &= u_1 x_1 + u_2 x_2 \\ &= \vec{u} \cdot \vec{x} \end{aligned}$$

- We get to choose $\vec{u} = (u_1, u_2)^T$.

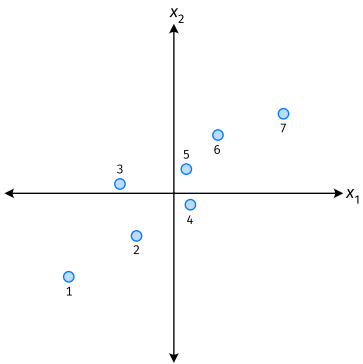
- Constraint: $\|\vec{u}\| = 1$.

Geometrically



- $\vec{u}$ defines a direction in $\mathbb{R}^2$.
- z is the projection of $\vec{x}$ onto that direction.
- Which direction should we pick?
  - Concluded: direction of max variance.

Another View



- Our data came to us in the standard basis.
- If we could pick a better basis, what would be our first basis vector?

Our Algorithm (Informally)

- **Given**: centered data $\vec{x}^{(1)}, \ldots, \vec{x}^{(n)} \in \mathbb{R}^d$
- Pick $\vec{u}$ to be the direction of "max variance"
- Create a new feature, z, for each point:

$$z^{(i)} = \vec{x}^{(i)} \cdot \vec{u}$$

PCA

- This algorithm is called **Principal Component Analysis**, or **PCA**.
- The direction of maximum variance is called the **principal component**.

Exercise

Suppose the direction of maximum variance in a centered data set is

$$\vec{u} = (1/\sqrt{2}, -1/\sqrt{2})^T$$

Let $\vec{x}^{(1)} = (3, -2)^T$ and $\vec{x}^{(2)} = (1, 4)^T$.
What are $z^{(1)}$ and $z^{(2)}$?

A) $z^{(1)} = \frac{1}{\sqrt{2}}, \quad z^{(2)} = \frac{-3}{\sqrt{2}}$

B) $z^{(1)} = \frac{5}{\sqrt{2}}, \quad z^{(2)} = \frac{3}{\sqrt{2}}$

C) $z^{(1)} = \frac{5}{\sqrt{2}}, \quad z^{(2)} = \frac{-3}{\sqrt{2}}$

Problem

- How do we compute the “direction of maximum variance”?

DSC 140B

Representation Learning

Lecture 06 | Part 2

Covariance Matrices

Variance

- We know how to compute the variance of a set of numbers $X = \{x^{(1)}, \ldots, x^{(n)}\}$:

$$\text{Var}(X) = \frac{1}{n} \sum_{i=1}^{n} (x^{(i)} - \mu)^2$$

- The variance measures the “spread” of the data

Generalizing Variance

- If we have two features, x_1 and x_2, we can compute the variance of each as usual:

$$\text{Var}(x_1) = \frac{1}{n}\sum_{i=1}^{n}(\vec{x}_1^{(i)} - \mu_1)^2$$

$$\text{Var}(x_2) = \frac{1}{n}\sum_{i=1}^{n}(\vec{x}_2^{(i)} - \mu_2)^2$$

- Can also measure how x_1 and x_2 "vary together".

Measuring Similar Information

- Features which share information if they *vary together*.
  - A.k.a., they “co-vary”
- Positive association: when one is above average, so is the other
- Negative association: when one is above average, the other is below average

Examples

- Positive: temperature and ice cream cones sold.
- Positive: temperature and shark attacks.
- Negative: temperature and coats sold.

Quantifying Co-Variance

- One approach is as follows:

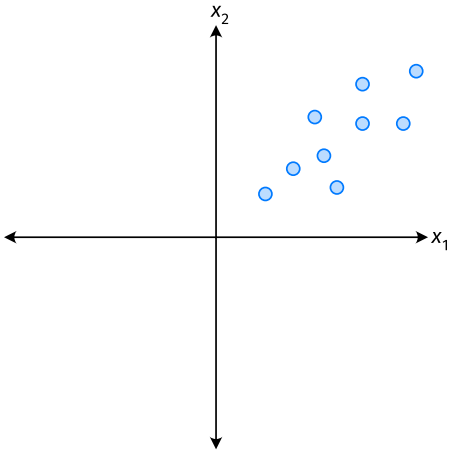
$$\text{Cov}(x_i, x_j) = \frac{1}{n} \sum_{k=1}^{n} (\vec{x}_i^{(k)} - \mu_i)(\vec{x}_j^{(k)} - \mu_j)$$

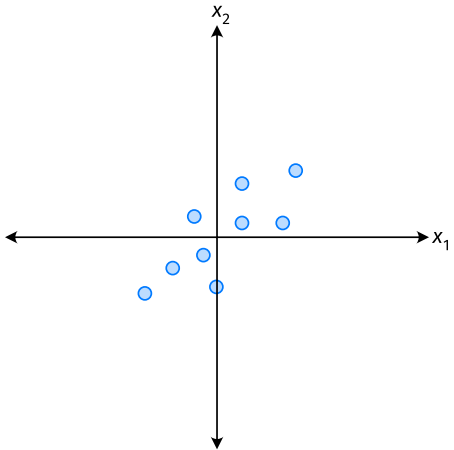
  - For each data point, multiply the value of feature *i* and feature *j*, then average these products.

  - This is the **covariance** of features *i* and *j*.

Centering

- We often **center** the data.





Centering

- Compute the mean of each feature:

$$\mu_j = \frac{1}{n} \sum_{1}^{n} \vec{x}_j^{(i)}$$

- Define new centered data:

$$\begin{pmatrix} \vec{x}_1^{(i)} \\ \vec{x}_2^{(i)} \\ \vdots \\ \vec{x}_d^{(i)} \end{pmatrix} \mapsto \begin{pmatrix} \vec{x}_1^{(i)} - \mu_1 \\ \vec{x}_2^{(i)} - \mu_2 \\ \vdots \\ \vec{x}_d^{(i)} - \mu_d \end{pmatrix}$$

Centering (Equivalently)

- Compute the mean of all data points:

$$\vec{\mu} = \frac{1}{n} \sum_{1}^{n} \vec{x}^{(i)}$$

- Define new centered data:

$$\vec{x}^{(i)} \mapsto \vec{x}^{(i)} - \vec{\mu}$$

Exercise

Center the data set:

$$\vec{x}^{(1)} = (1, 2, 3)^T$$
$$\vec{x}^{(2)} = (-1, -1, 0)^T$$
$$\vec{x}^{(3)} = (0, 2, 3)^T$$

Covariance (Again)

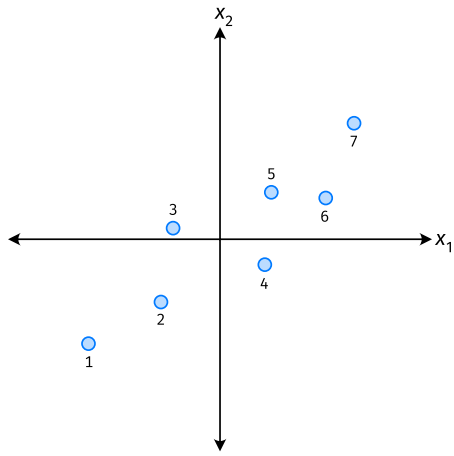
- If the data are **centered**, covariance is:

$$\text{Cov}(x_i, x_j) = \frac{1}{n} \sum_{k=1}^{n} \vec{x}_i^{(k)} \vec{x}_j^{(k)}$$

Quantifying Covariance

- Assume the data are **centered**.

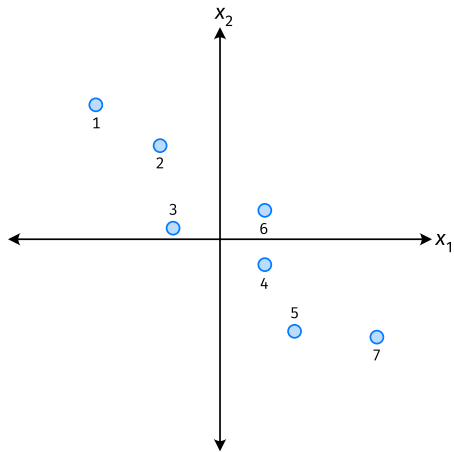
$$\text{Covariance} = \frac{1}{7}\sum_{i=1}^{7} \vec{x}_1^{(i)} \times \vec{x}_2^{(i)}$$



Quantifying Covariance

- Assume the data are **centered**.

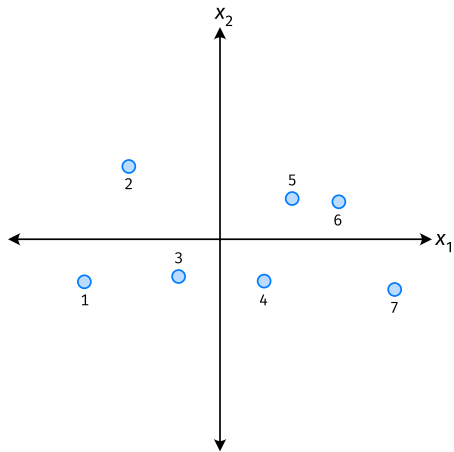
$$\text{Covariance} = \frac{1}{7}\sum_{i=1}^{7} \vec{x}_1^{(i)} \times \vec{x}_2^{(i)}$$



Quantifying Covariance

- Assume the data are **centered**.

$$\text{Covariance} = \frac{1}{7}\sum_{i=1}^{7} \vec{x}_1^{(i)} \times \vec{x}_2^{(i)}$$



Quantifying Covariance

- The **covariance** quantifies extent to which two variables “vary together”.
- Assume we have centered the data.
- The **sample covariance** of feature i and j is:

$$\sigma_{ij} = \frac{1}{n} \sum_{k=1}^{n} \vec{x}_i^{(k)} \vec{x}_j^{(k)}$$

Exercise

True or False: $\sigma_{ij} = \sigma_{ji}$?

$$\sigma_{ij} = \frac{1}{n} \sum_{k=1}^{n} \vec{x}_i^{(k)} \vec{x}_j^{(k)}$$

Covariance Matrices

- Given data $\vec{x}^{(1)}, \ldots, \vec{x}^{(n)} \in \mathbb{R}^d$.
- The **sample covariance matrix** C is the $d \times d$ matrix whose ij entry is defined to be σ_{ij}.

$$\sigma_{ij} = \frac{1}{n} \sum_{k=1}^{n} \vec{x}_i^{(k)} \vec{x}_j^{(k)}$$

Observations

- Diagonal entries of C are the variances.
- The matrix is **symmetric**!

Note

- Sometimes you'll see the sample covariance defined with $1/(n-1)$ instead of $1/n$:

$$\sigma_{ij} = \frac{1}{n-1} \sum_{k=1}^{n} \vec{x}_i^{(k)} \vec{x}_j^{(k)}$$

- This is an **unbiased** estimator of the population covariance.
- Our definition is the **maximum likelihood** estimator.
- In practice, it doesn't matter: $1/(n-1) \approx 1/n$.
- For consistency, in this class use $1/n$.

Exercise

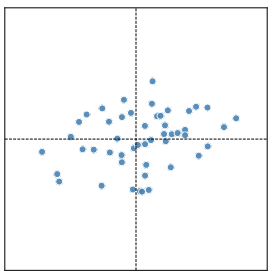
Which of the following could be the covariance matrix for the data shown below?

A) $\begin{pmatrix} 4 & 2 \\ 2 & 2 \end{pmatrix}$

B) $\begin{pmatrix} 4 & -2 \\ -2 & 2 \end{pmatrix}$

C) $\begin{pmatrix} 2 & 2 \\ 2 & 4 \end{pmatrix}$

D) $\begin{pmatrix} 4 & 0 \\ 0 & 2 \end{pmatrix}$



Computing Covariance

- There is a “trick” for computing sample covariance matrices.
- Step 1: make $n \times d$ data matrix, X
- Step 2: make Z by centering columns of X
- Step 3: $C = \frac{1}{n} Z^T Z$

Computing Covariance (in code)[1]

```
>>> mu = X.mean(axis=0)
>>> Z = X - mu
>>> C = 1 / len(X) * Z.T @ Z
```

[1] Or use `np.cov`

Meaning of the Covariance Matrix

- On the one hand, C is just a table of numbers.
- But remember: every matrix represents a linear transformation.
- What linear transformation does C represent?

Meaning of the Covariance Matrix

- Suppose $\vec{u}$ is a unit vector listing our "mixture coefficients":

$$z = u_1 x_1 + u_2 x_2 + \cdots + u_d x_d$$

- $C\vec{u}$ computes the covariances of the new feature z with each of the original features, $x_1, \ldots, x_d$:

$$C\vec{u} = \left(\text{Cov}(z, x_1), \text{Cov}(z, x_2), \ldots, \text{Cov}(z, x_d)\right)^T$$

- We'd like each to be large.
  - Then, the new feature would be highly correlated with the original features.

Intuition

- $\|C\vec{u}\|$ is large when the new feature $z = \vec{u} \cdot \vec{x}$ is **highly correlated** with the original features.

Intuition

- $\|C\vec{u}\|$ is large when the new feature $z = \vec{u} \cdot \vec{x}$ is **highly correlated** with the original features.
- That is, when z contains a lot of the same information.

Intuition

- $\|C\vec{u}\|$ is large when the new feature $z = \vec{u} \cdot \vec{x}$ is **highly correlated** with the original features.
- That is, when z contains a lot of the same information.
- To maximize this correlation, we want to find $\vec{u}$ which maximizes $\|C\vec{u}\|$.

DSC 140B

Representation Learning

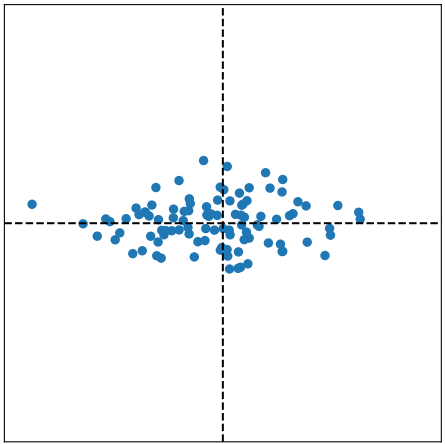
Lecture 06 | Part 3

Visualizing Covariance Matrices

Visualizing Covariance Matrices

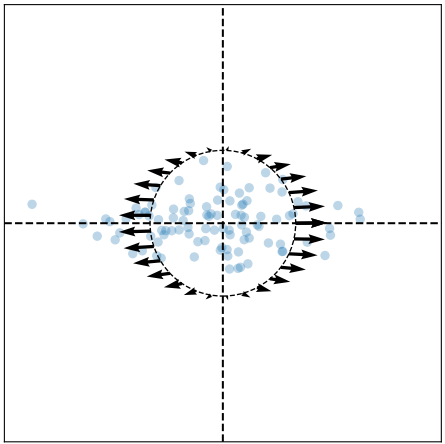
- Covariance matrices are symmetric.
- They have axes of symmetry (eigenvectors and eigenvalues).
- What are they?

Visualizing Covariance Matrices



$$C \approx \begin{pmatrix} & \\ & \end{pmatrix}$$

Visualizing Covariance Matrices

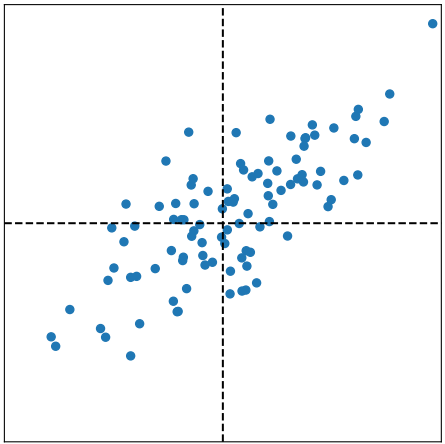


Eigenvectors:

$\vec{u}^{(1)} \approx$

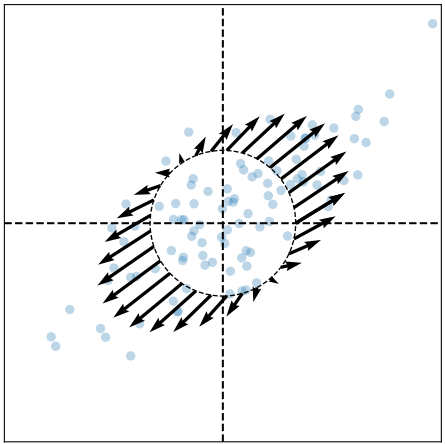
$\vec{u}^{(2)} \approx$

Visualizing Covariance Matrices



$$C \approx \begin{pmatrix} & \\ & \end{pmatrix}$$

Visualizing Covariance Matrices



Eigenvectors:

$\vec{u}^{(1)} \approx$

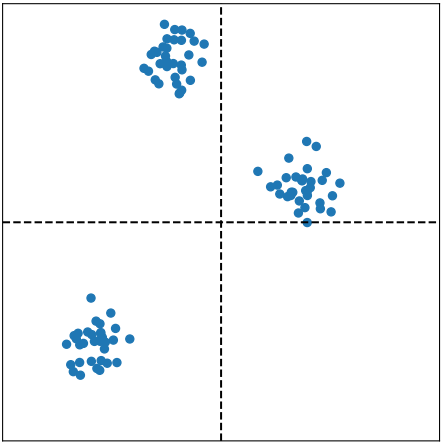
$\vec{u}^{(2)} \approx$

Observations

- The **eigenvectors** of the covariance matrix describe the data's "principal directions"
  - C tells us something about data's shape.
- The **top eigenvector** points in the direction of "maximum variance".
- The **top eigenvalue** is proportional to the variance in this direction.

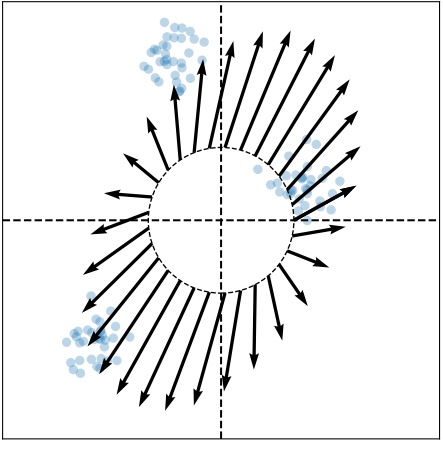
Caution

- The data doesn't always look like this.
- We can always compute covariance matrices.
- They just may not describe the data's shape very well.



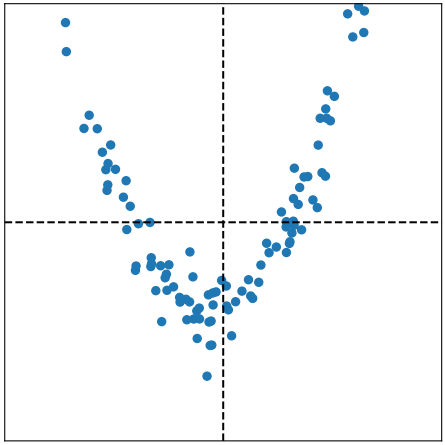
Caution

- The data doesn't always look like this.
- We can always compute covariance matrices.
- They just may not describe the data's shape very well.



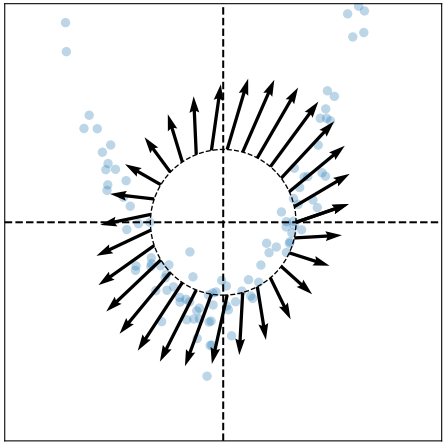
Caution

- The data doesn't always look like this.
- We can always compute covariance matrices.
- They just may not describe the data's shape very well.



Caution

- The data doesn't always look like this.
- We can always compute covariance matrices.
- They just may not describe the data's shape very well.



DSC 140B

Representation Learning

Lecture 06 | Part 4

PCA, More Formally

The Story (So Far)

- We want to create a single new feature, z.
- Our idea: $z = \vec{x} \cdot \vec{u}$; choose $\vec{u}$ to point in the "direction of maximum variance".
- Intuition: the top eigenvector of the covariance matrix points in direction of maximum variance.

More Formally...

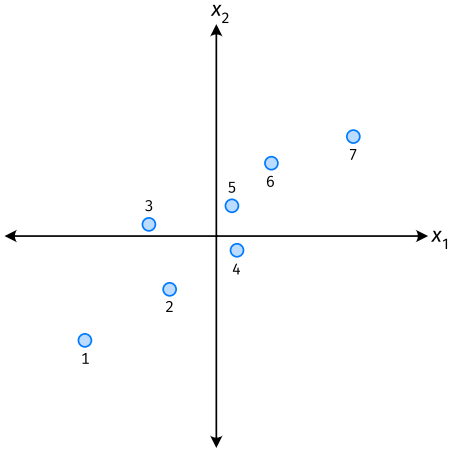
- We haven't actually defined "direction of maximum variance"
- Let's derive PCA more formally.

Variance in a Direction

- Given centered data $\vec{x}^{(1)}, \ldots, \vec{x}^{(n)}$ and a unit vector $\vec{u}$.
- $z^{(i)} = \vec{x}^{(i)} \cdot \vec{u}$ is the new feature for $\vec{x}^{(i)}$.
- The **variance in the direction of $\vec{u}$** is defined to be the variance of the new features:

$$\begin{aligned} \text{Var}(z) &= \frac{1}{n} \sum_{i=1}^{n} (z^{(i)} - \mu_z)^2 \\ &= \frac{1}{n} \sum_{i=1}^{n} \left(\vec{x}^{(i)} \cdot \vec{u} - \mu_z \right)^2 \end{aligned}$$

Example



Note

- If the data are centered, then $\mu_z = 0$ and the variance of the new features is:

$$\begin{aligned}\text{Var}(z) &= \frac{1}{n}\sum_{i=1}^{n}(z^{(i)})^2 \\ &= \frac{1}{n}\sum_{i=1}^{n}\left(\vec{x}^{(i)} \cdot \vec{u}\right)^2\end{aligned}$$

Goal

- The variance of a data set in the direction of $\vec{u}$ is:

$$g(\vec{u}) = \frac{1}{n} \sum_{i=1}^{n} \left(\vec{x}^{(i)} \cdot \vec{u} \right)^2$$

- Our goal: Find a unit vector $\vec{u}$ which maximizes g.

Claim

$$\frac{1}{n}\sum_{i=1}^{n}\left(\vec{x}^{(i)} \cdot \vec{u}\right)^2 = \vec{u}^T C \vec{u}$$

- Proven on this week's homework.

Our Goal (Again)

- Find a unit vector $\vec{u}$ which maximizes $\vec{u}^T C \vec{u}$.

Recall

- When C is symmetric, the unit vector which maximizes the quadratic form $\vec{u}^T C \vec{u}$ is the eigenvector of C with the largest eigenvalue.

- **Solution**: the direction of maximum variance is the top eigenvector of the covariance matrix.

PCA (for a single new feature)

- **Given**: centered data $\vec{x}^{(1)}, \ldots, \vec{x}^{(n)} \in \mathbb{R}^d$

1. Compute the covariance matrix, C.
2. Compute the top[2] eigenvector $\vec{u}$, of C.
3. For $i \in \{1, \ldots, n\}$, create new feature:

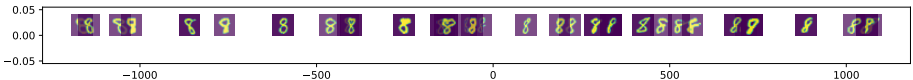
$$z^{(i)} = \vec{u} \cdot \vec{x}^{(i)}$$

[2] All eigenvalues are positive. Why?

A Parting Example

- MNIST: 60,000 images in 784 dimensions
- Principal component: $\vec{u} \in \mathbb{R}^{784}$
- We can project an image in $\mathbb{R}^{784}$ onto $\vec{u}$ to get a single number representing the image

Example



DSC 140B

Representation Learning

Lecture 06 | Part 5

Dimensionality Reduction with d ≥ 2

So far: PCA

- **Given**: centered data $\vec{x}^{(1)}, \ldots, \vec{x}^{(n)} \in \mathbb{R}^d$
- **Map**: each data point $\vec{x}^{(i)}$ to a single feature, z_i.
  - Idea: maximize the variance of the new feature
- **PCA**: Let $z_i = \vec{x}^{(i)} \cdot \vec{u}$, where $\vec{u}$ is top eigenvector of covariance matrix, C.

Now: More PCA

- **Given**: centered data $\vec{x}^{(1)}, \ldots, \vec{x}^{(n)} \in \mathbb{R}^d$
- **Map**: each data point $\vec{x}^{(i)}$ to k new features, $\vec{z}^{(i)} = (z_1^{(i)}, \ldots, z_k^{(i)})$.

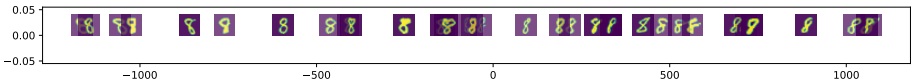
A Single Principal Component

- Recall: the **principal component** is the top eigenvector $\vec{u}$ of the covariance matrix, C
- It is a unit vector in $\mathbb{R}^d$
- Make a new feature $z \in \mathbb{R}$ for point $\vec{x} \in \mathbb{R}^d$ by computing $z = \vec{x} \cdot \vec{u}$
- This is dimensionality reduction from $\mathbb{R}^d \to \mathbb{R}^1$

Example

- MNIST: 60,000 images in 784 dimensions
- Principal component: $\vec{u} \in \mathbb{R}^{784}$
- We can project an image in $\mathbb{R}^{784}$ onto $\vec{u}$ to get a single number representing the image

Example



Another Feature?

- Clearly, mapping from $\mathbb{R}^{784} \rightarrow \mathbb{R}^1$ loses a lot of information
- What about mapping from $\mathbb{R}^{784} \rightarrow \mathbb{R}^2$? $\mathbb{R}^k$?

A Second Feature

- Our first feature is a mixture of features, with weights given by unit vector $\vec{u}^{(1)} = (u_1^{(1)}, u_2^{(1)}, \ldots, u_d^{(1)})^T$.

$$z_1 = \vec{u}^{(1)} \cdot \vec{x} = u_1^{(1)} x_1 + \ldots + u_d^{(1)} x_d$$

- To maximize variance, choose $\vec{u}^{(1)}$ to be top eigenvector of C.

A Second Feature

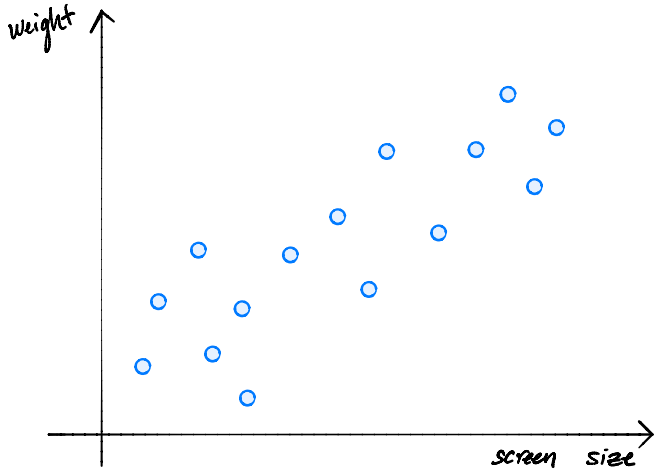
- Make same assumption for second feature:

$$z_2 = \vec{u}^{(2)} \cdot \vec{x} = u_1^{(2)} x_1 + \ldots + u_d^{(2)} x_d$$

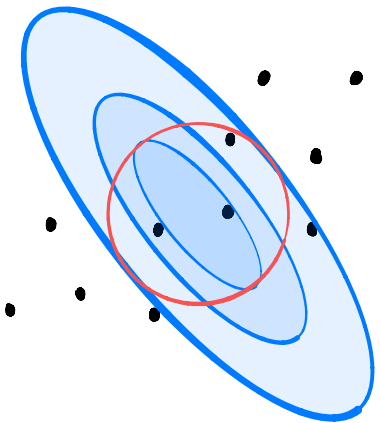
- How do we choose $\vec{u}^{(2)}$?

- We should choose $\vec{u}^{(2)}$ to be **orthogonal** to $\vec{u}^{(1)}$.
  - No “redundancy”.

A Second Feature



A Second Feature



Intuition

- Claim: if $\vec{u}$ and $\vec{v}$ are eigenvectors of a symmetric matrix with distinct eigenvalues, they are orthogonal.
- We should choose $\vec{u}^{(2)}$ to be an **eigenvector** of the covariance matrix, C.
- The second eigenvector of C is called the **second principal component**.

A Second Principal Component

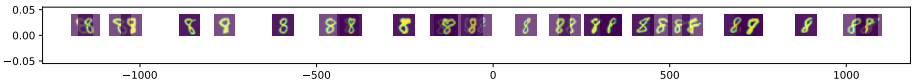
- Given a covariance matrix C.
- The principal component $\vec{u}^{(1)}$ is the top eigenvector of C.
  - Points in the direction of maximum variance.
- The *second* principal component $\vec{u}^{(2)}$ is the *second* eigenvector of C.
  - Out of all vectors orthogonal to the principal component, points in the direction of max variance.

PCA: Two Components

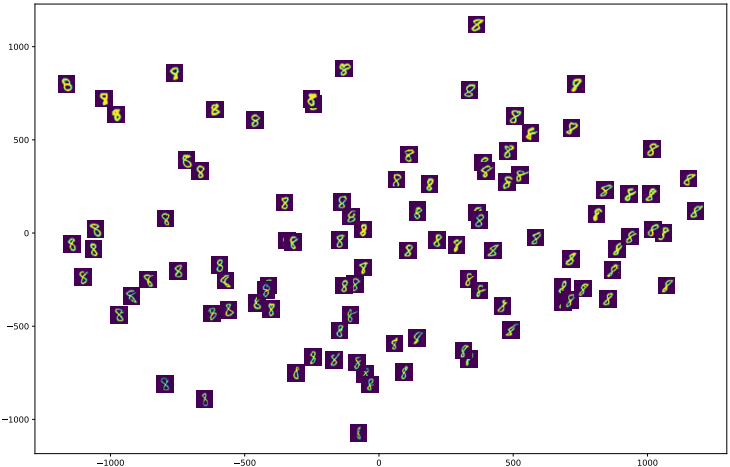
- Given centered data $\{\vec{x}^{(1)}, ..., \vec{x}^{(n)}\} \in \mathbb{R}^d$.
- Compute covariance matrix C, top two eigenvectors $\vec{u}^{(1)}$ and $\vec{u}^{(2)}$.
- For any vector $\vec{x} \in \mathbb{R}$, its new representation in $\mathbb{R}^2$ is $\vec{z} = (z_1, z_2)^T$, where:

$$z_1 = \vec{x} \cdot \vec{u}^{(1)}$$
$$z_2 = \vec{x} \cdot \vec{u}^{(2)}$$

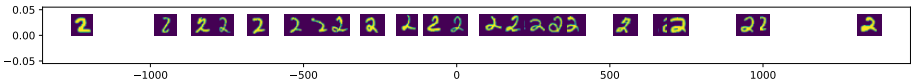
Example



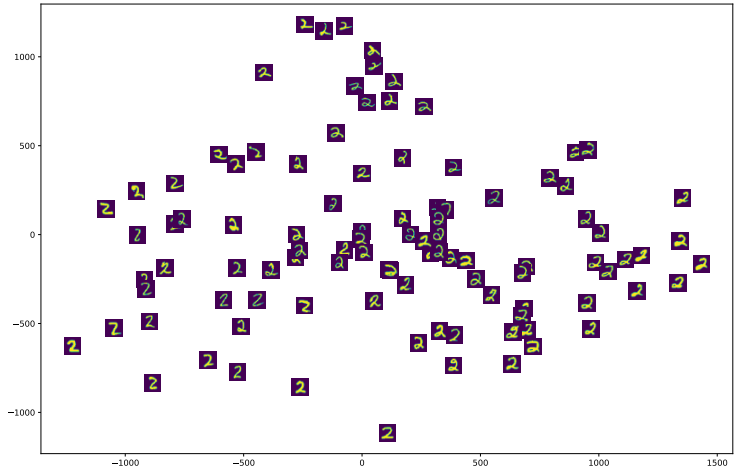
Example



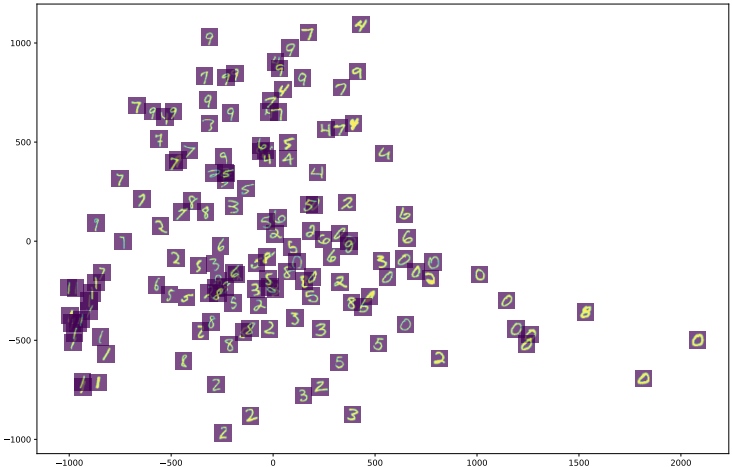
Example



Example



Example



PCA: k Components

- Given centered data $\{\vec{x}^{(1)}, ..., \vec{x}^{(n)}\} \in \mathbb{R}^d$, number of components k.
- Compute covariance matrix C, top $k \leq d$ eigenvectors $\vec{u}^{(1)}$, $\vec{u}^{(2)}$, ..., $\vec{u}^{(k)}$.
- For any vector $\vec{x} \in \mathbb{R}$, its new representation in $\mathbb{R}^k$ is $\vec{z} = (z_1, z_2, ... z_k)^T$, where:

$$
\begin{aligned}
z_1 &= \vec{x} \cdot \vec{u}^{(1)} \\
z_2 &= \vec{x} \cdot \vec{u}^{(2)} \\
&\vdots \\
z_k &= \vec{x} \cdot \vec{u}^{(k)}
\end{aligned}
$$

Matrix Formulation

- Let X be the **centered data matrix** (n rows, d columns)
- Let U be matrix of the top k eigenvectors as columns (d rows, k columns)
- The new representation: $Z = XU$

DSC 140B

Representation Learning

Lecture 06 | Part 6

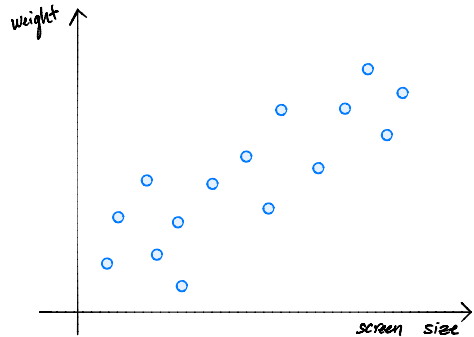
Reconstructions

Reconstructing Points

- PCA helps us reduce dimensionality from $\mathbb{R}^d \rightarrow R^k$
- Suppose we have the “new” representation in $\mathbb{R}^k$.
- Can we “go back” to $\mathbb{R}^d$?
- And why would we want to?

Back to $\mathbb{R}^d$

- Suppose new representation of $\vec{x}$ is z.
- $z = \vec{x} \cdot \vec{u}^{(1)}$
- Idea: $\vec{x} \approx z\vec{u}^{(1)}$



Reconstructions

- Given a "new" representation of $\vec{x}$, $\vec{z} = (z_1, \ldots, z_k) \in \mathbb{R}^k$
- And top k eigenvectors, $\vec{u}^{(1)}, \ldots, \vec{u}^{(k)}$
- The **reconstruction** of $\vec{x}$ is

$$z_1\vec{u}^{(1)} + z_2\vec{u}^{(2)} + \ldots + z_k\vec{u}^{(k)} = U\vec{z}$$

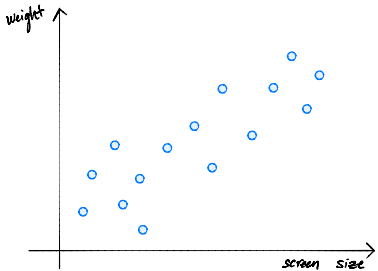
Reconstruction Error

- The reconstruction *approximates* the original point, $\vec{x}$.
- The **reconstruction error** for a single point, $\vec{x}$:

$$\|\vec{x} - U\vec{z}\|^2$$

- Total reconstruction error:

$$\sum_{i=1}^{n} \|\vec{x}^{(i)} - U\vec{z}^{(i)}\|^2$$



DSC 140B

Representation Learning

Lecture 06 | Part 7

Interpreting PCA

Three Interpretations

- What is PCA doing?
- PCA is an **orthogonal projection** of the data onto a lower-dimensional subspace.
- Three interpretations:
  1. Maximizing total variance
  2. Finding the best reconstruction
  3. Decorrelating features

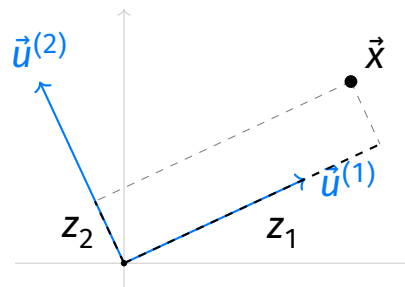
Orthogonal Projections

- **Recall:** we change basis by projecting points onto new basis vectors.

$$z_1 = \vec{x} \cdot \vec{u}^{(1)} \quad , \quad z_2 = \vec{x} \cdot \vec{u}^{(2)}$$

- If U is the matrix with columns $\vec{u}^{(1)}, \vec{u}^{(2)}$, and X is the data matrix, then the new representation is:

$$Z = XU$$



Orthogonal Projections

- If X is $n \times d$ and we use d basis vectors, then XU just changes basis.
  - Still d dimensions, no information lost.

- But if we use $k < d$ basis vectors, then XU projects points onto a k-dimensional subspace.
  - Some information is lost.
  - This is dimensionality reduction, and it is what PCA does.

Visualization

http://dsc140b.com/static/vis/pca-3d-spheroid/

PCA

- Any choice of k orthonormal basis vectors defines a projection onto a k-dimensional subspace.
- PCA chooses the basis vectors to be the top k eigenvectors of the covariance matrix C.
- In a sense, PCA is the **best** choice of basis.

View #1: Maximizing Total Variance

- The basis chosen by PCA maximizes the “total variance” of the new data.

Deriving PCA

- We derived PCA by maximizing variance.
- **Idea**: the direction of max variance is most interesting; let's use it as our first basis vector.
- We found that the direction of max variance is the top eigenvector of the covariance matrix C.

Deriving PCA

- We then iterated.
- **Idea**: the *second* most interesting direction is the one with max variance *orthogonal* to the first.
- We found that this is the *second* eigenvector of C.

Deriving PCA

- **Idea**: the k most interesting directions are the k directions with maximum variance, orthogonal to each other.
- We found that these are the top k eigenvectors of C.
- They capture the “shape” of the data.

Maximizing Variance

- At each step, we chose the direction that maximized variance.
- Overall, PCA chooses the k-dimensional subspace that maximizes the “total variance”.

Total Variance

- Consider a data set $\vec{x}^{(1)}, \ldots, \vec{x}^{(n)} \in \mathbb{R}^d$ with features $x_1, x_2, \ldots, x_d$.

- The **total variance** of the data is

$$\sum_{j=1}^{d} \text{Var}(\text{feature } x_j)$$

- Example: with phone width, height as features, the total variance is:

$$\text{Var}(\text{widths}) + \text{Var}(\text{heights})$$

Total Variance

- Now say you do an orthogonal projection of the data onto a k-dimensional subspace: $Z = XU$.
- This is a new data set with new features.
- The **total variance** of the new data set is

$$\sum_{j=1}^{k} \text{Var}(\text{feature } z_j)$$

Claim

- Out of all orthogonal projections onto k-dimensional subspaces, PCA **maximizes** the total variance of the new data.

Visualization

http://dsc140b.com/static/vis/pca-3d-spheroid/

Variance and Eigenvalues

- **Recall:** variance in direction of unit vector $\vec{u}$ is

$$\text{Var}(\vec{u}) = \vec{u}^T C \vec{u}$$

- If $\vec{u}$ is an eigenvector of C with eigenvalue λ, then

$$\text{Var}(\vec{u}) = \vec{u}^T C \vec{u} = \vec{u}^T (\lambda \vec{u}) = \lambda (\vec{u}^T \vec{u}) = \lambda$$

- Thus, the variance of the kth new PCA feature is the kth largest eigenvalue λ_k.

Total Variance of PCA Features

- If we use PCA to project onto k dimensions, the total variance of the new data is:

$$\sum_{j=1}^{k} \text{Var}(\text{feature } z_j) = \sum_{j=1}^{k} \lambda_j$$

- I.e., the sum of the top k eigenvalues of C.

Exercise

Imagine we throw a perfectly thin chocolate chip pancake in the air.

While it is up there, you measure the 3D coordinates of n chocolate chips embedded in the pancake and compute the sample covariance matrix of the data.

What is the third eigenvalue of the covariance matrix?

Main Idea

PCA maximizes the total variance of the new data. I.e., chooses the most “interesting” new features which are not redundant.

View #2: Minimizing Reconstruction Error

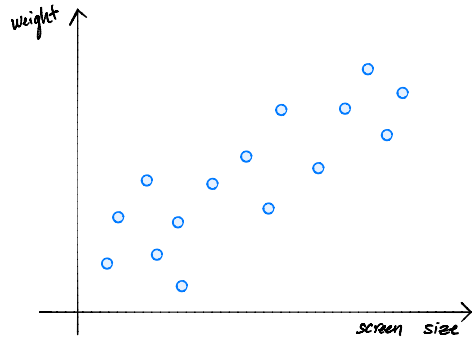
- PCA can also be viewed as minimizing the “reconstruction error”.

Reconstructing Points

- PCA reduces dimensionality from $\mathbb{R}^d \to \mathbb{R}^k$
- Suppose we have the “new” representation in $\mathbb{R}^k$.
- Can we “reconstruct” the original point in $\mathbb{R}^d$?

Back to $\mathbb{R}^d$

- Suppose new representation of $\vec{x}$ is z.
- $z = \vec{x} \cdot \vec{u}^{(1)}$
- Idea: $\vec{x} \approx z\vec{u}^{(1)}$



Reconstructions

- Suppose we map $\vec{x} \in \mathbb{R}^d$ to $\vec{z} \in \mathbb{R}^k$ by projecting onto k orthonormal basis vectors $\vec{u}^{(1)}, \ldots, \vec{u}^{(k)}$:

$$z_1 = \vec{x} \cdot \vec{u}^{(1)} \quad , \quad z_2 = \vec{x} \cdot \vec{u}^{(2)} \quad , \quad \ldots \quad , \quad z_k = \vec{x} \cdot \vec{u}^{(k)}$$

- Equivalently, $\vec{z} = U^T \vec{x}$ where U is the $d \times k$ matrix with columns $\vec{u}^{(1)}, \ldots, \vec{u}^{(k)}$.

- The **reconstruction** of $\vec{x}$ is

$$z_1 \vec{u}^{(1)} + z_2 \vec{u}^{(2)} + \ldots + z_k \vec{u}^{(k)} = U\vec{z} = UU^T \vec{x}$$

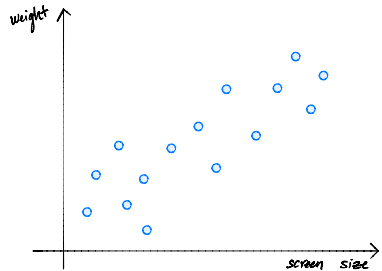
Reconstruction Error

- The reconstruction $UU^T\vec{x}$ *approximates* the original point, $\vec{x}$.
- The **reconstruction error** for a single point, $\vec{x}$:

$$\|\vec{x} - U\vec{U}^T\vec{x}\|^2$$

- Total reconstruction error:

$$\sum_{i=1}^{n} \|\vec{x}^{(i)} - UU^T\vec{x}^{(i)}\|^2$$



Goal

- Here's another way to frame dimensionality reduction.
- **Given**: centered data $\vec{x}^{(1)}, \ldots, \vec{x}^{(n)} \in \mathbb{R}^d$, dimensionality k.
- **Goal**: find orthonormal $d \times k$ projection matrix U that **minimizes** total reconstruction error:

$$\sum_{i=1}^{n} \| \vec{x}^{(i)} - U\vec{z}^{(i)} \|^2$$

Visualization

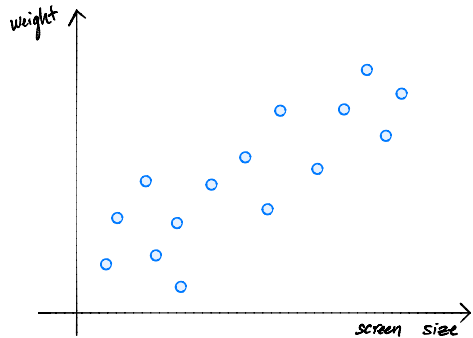
http://dsc140b.com/static/vis/pca-max_variance/

View #2: Reconstruction Error

- Claim: PCA **minimizes** the total reconstruction error out of all orthogonal projections onto k-dimensional subspaces.

Lines of Best Fit

- Both **least squares regression** and **PCA** find "lines/planes of best fit".
- They differ in how they measure error.
- **Least squares**: vertical distance to line/plane.
- **PCA**: perpendicular distance to line/plane.



Exercise

The chocolate chip pancake is back in the air!

We quickly perform PCA on the 3D coordinates of the chocolate chips to reduce the data to 2D.

What is the reconstruction error?

Main Idea

PCA minimizes the reconstruction error. It is the "best" projection of points onto a linear subspace of dimensionality k. When $k = d$, the reconstruction error is zero.

View #3: Decorrelation

- PCA has the effect of “decorrelating” the features.

Exercise

Suppose we perform PCA to get new features $z_1, z_2, \ldots, z_k$ and we compute the sample covariance matrix of the new data.

True or False: the covariance matrix will be diagonal.

Covariance of PCA Features

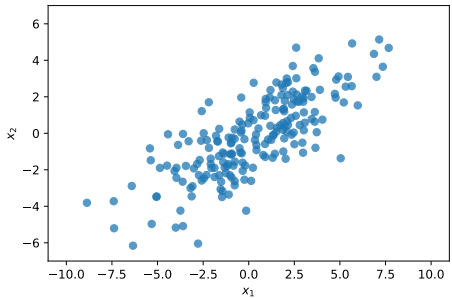
- The covariance matrix of the new features is diagonal:

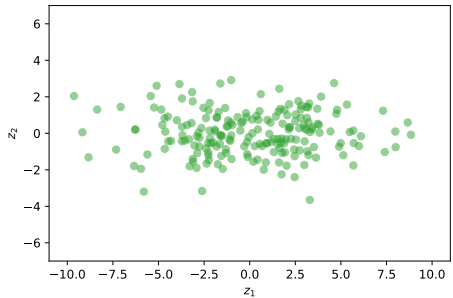
$$\begin{pmatrix} \lambda_1 & 0 & 0 & \cdots & 0 \\ 0 & \lambda_2 & 0 & \cdots & 0 \\ 0 & 0 & \lambda_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \lambda_k \end{pmatrix}$$

- This is because we have changed to the eigenbasis of the covariance matrix.

Decorrelated Features

- A diagonal covariance matrix means the features are **uncorrelated**.





Main Idea

PCA learns a new representation by rotating the data into a basis where the features are uncorrelated (not redundant).

That is: the natural basis vectors are the principal directions (eigenvectors of the covariance matrix). PCA changes the basis to this natural basis.

DSC 140B

Representation Learning

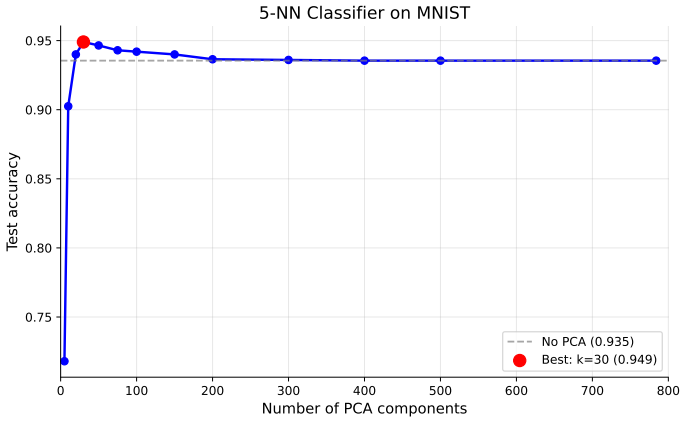
Lecture 06 | Part 8

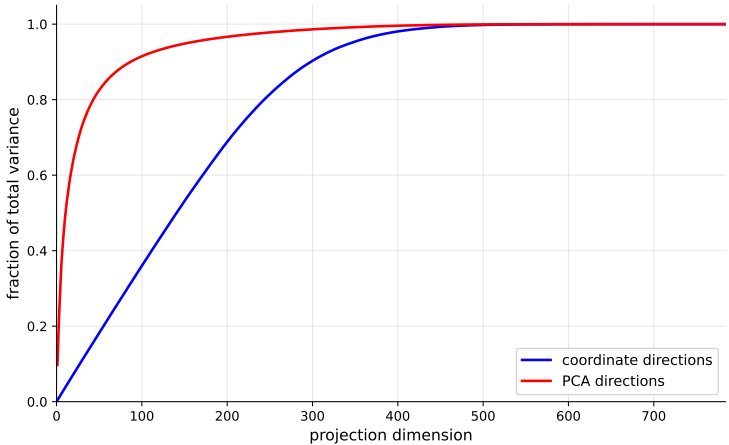
PCA in Practice

PCA in Practice

- PCA is often used in **preprocessing** before classifier is trained, etc.
- Must choose number of dimensions, k.
- One way: cross-validation.
- Another way: the elbow method.

Example: PCA Before k-NN Classifier




1.0
0.8
0.6
0.4
0.2
0.0
fraction of total variance
0
100
200
300
400
500
600
700
projection dimension
coordinate directions
PCA directions

Caution

- PCA's assumption: variance is interesting
- PCA is totally unsupervised
- The direction most meaningful for classification may not have large variance!

Example